

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII  
MOLDOVA**

**UNIVERSITATEA DE STAT „ALECU RUSSO” DIN BĂLȚI  
FACULTATEA DE ȘTIINȚE REALE, ECONOMICE ȘI ALE MEDIULUI  
CATEDRA DE MATEMATICĂ ȘI INFORMATICĂ**

## **CURRICULUM**

**la unitatea de curs**

## **INFORMATICA II**

**Ciclul I, studii superioare de licență**

**Codul și denumirea domeniului general de studiu: 071 Inginerie și activități inginerești**

**Codul și denumirea domeniului de formare profesională: 0710 Inginerie și management**

**Codul și denumirea specialității:**

**0710.1 Inginerie și management pe ramuri (în transport auto)**

**Forma de învățământ: cu frecvență la zi**

**Autori:**

**conf. universitar, dr. Mircea PETIC**



**lector universitar Vitalie ȚICĂU**

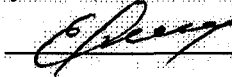


**BĂLȚI, 2018**

Discutat și aprobat la ședința Catedrei de matematică și informatică

Procesul-verbal nr. 8 din 03.04.2018

Șeful catedrei de matematică și informatică


 conf. univ., dr. Eugeniu PLOHOTNIUC

Discutat și aprobat la ședința Consiliului Facultății de Științe Reale, Economice și ale Mediului

Procesul verbal nr. 41 din 10.05.2018

Decanul Facultății de Științe Reale, Economice și ale Mediului,



 conf. univ., dr. Ina CIOBANU

© Mircea Petic, USARB, 2018

© Vitalie ȚICĂU, USARB, 2018

**Informații de identificare a unității de curs****Facultatea:** Științe Reale, Economice și ale Mediului**Catedra:** Matematică și informatică**Domeniul general de studiu:** 071 Inginerie și activități ingineresti**Domeniul de formare profesională la ciclul I:** 0710 Inginerie și management**Denumirea specializării:** 0710.1 Inginerie și management pe ramuri (în transport auto)**Denumirea unității de curs:** Informatica II**Administrarea unității de curs**

Codul unității de curs	Credite ECTS	Total ore	Repartizarea orelor				Forma de evaluare	Limba de predare
			Prelegeri	Seminar	Laborator	Lucrul ind.		
F.02.0.013	4	120	30	–	30	60	Examen	rom/rus

**Anul de studiu și semestrul în care se studiază:** Anul I, sem. 2.**Statutul:** Unitate de curs obligatorie fundamentală.**Informații referitoare la cadrele didactice**

**Mircea PETIC**, dr. în informatică, conferențiar universitar. Absolvent al Universității de Stat „A. Russo” din Bălți, specialitatea „Informatica și limba engleză aplicată”. A efectuat stagii în diverse universități, inclusiv peste hotare, unde s-a specializat în domeniul informaticii, utilizarea tehnologiei informației și a comunicațiilor în învățământ, procesarea limbajului natural, programarea obiect orientată, programarea logică și funcțională, ingineria programării, managementul proiectelor. E-mail: [petic.mircea@yahoo.com](mailto:petic.mircea@yahoo.com)

Orele de consultații - miercuri: 15.00 -17.00. Consultațiile se oferă atât în regim „față-în-față”, cât și prin utilizarea poștei electronice, YahooMessenger (petic.mircea) și Skype sau rețeaua socială facebook. Numele în Skype – mir\_cescu.

**Vitalie ȚICĂU**, lector superior universitar. Absolvent a Universității de Stat din Moldova, specialitatea „Matematica aplicată”. A finalizat studiile de doctorat la specialitatea „Matematica de calcul”. A realizat publicații aplicative și metodice cu tematica: „Structuri discrete”, „Limbaje de programe orientate obiect”, „Analiza numerică”, „Prelucrarea informației grafice”, „Rezolvări de probleme din domeniul informaticii”. Formator permanent din anul 2003 în cadrul cursurilor de formare continuă a profesorilor de informatică.

**Biroul:** Sala de calculatoare 140.**Catedra de matematică și informatică:** aula 145. Tel. 0 231 52 488.

**E-mail:** VitalieSTicau@gmail.com

**Orele de consultații:** marți, joi: 14:30 – 16:00. Consultațiile se oferă atât în regim „față-în-față”, cât și prin utilizarea poștei electronice.

### **Integrarea unității de curs în programul de studii**

Unitatea de curs „Informatica II” este fundamentală și are scop dezvoltarea gândirii logice, precum și dezvoltarea multilaterală a studenților. Unitatea de curs respectivă formează și dezvoltă competențe profesionale ale viitorului inginer, analizând specificul formării deprinderilor informatice, atât în aspect general, cât și particular pe domeniu.

Unitatea de curs „Informatica II” este o continuare a cursului fundamental „Informatica I” și contribuie la dezvoltarea competențelor de programare a studenților, aplicând , aplicând limbajul de programare C/C++.

Prin conținutul său și activitățile de învățare a studenților, unitatea de curs „Informatica II” contribuie la dezvoltarea mai multor competențe generice.

### **Competențe prealabile**

1. Utilizarea eficientă a resurselor sistemelor de calcul, de operare și ale Internetului;
2. Posedarea cunoștințelor de bază ale unității de curs „Informatica I”;
3. Posedarea la nivel teoretic și aplicativ a informaticii preuniversitare în limitele standardelor de studiu eficient al învățării;
4. Posedarea deprinderilor de elaborare și analiză a algoritmilor de rezolvare a problemelor practice în baza precizării problemei;
5. Posedarea deprinderilor de lucru cu MS OFFICE.

### **Competențele formate/dezvoltate în cadrul unității de curs**

Prin conținutul său și activitățile de învățare a studenților, unitatea de curs *Informatica II* contribuie la dezvoltarea diverselor competențe de programare. Aceste competențe servesc ca instrumente de formare a următoarelor competențe profesionale (CP) și a competențelor transversale (CT), vizate în planurile de învățământ la specialitatea nominalizată:

**CP1.2.** Utilizarea cunoștințelor de bază din disciplinele fundamentale pentru explicarea detaliată și interpretarea rezultatelor teoretice, fenomenelor sau proceselor specifice ingineriei și managementului.

**CP3.3.** Aplicarea de principii și metode de bază din tehnologiile digitale pentru modelarea, proiectarea, asistată de calculator a tehnologiilor, produselor și proceselor automate și neautomate specifice ingineriei.

**CP4.4.** Utilizarea adecvată de criterii și metode standard de evaluare, pentru a aprecia calitatea, avantajele și limitele proceselor de programare, dezvoltare și mentenanță ale aplicațiilor informatice în limbaje de nivel înalt.

**CT1.** Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unei atitudini responsabile față de domeniul științific și didactic, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională.

### Finalitățile cursului

La finalizarea studierii unității de curs studentul va fi capabil:

- să proiecteze algoritmul și etapele de elaborare și rezolvare a problemei.
- să rezolve probleme concrete, bazate pe aplicarea tipurilor de date omogene și mixte.
- să rezolve probleme concrete evidențiind aplicarea structurilor dinamice/statice de date.
- să traseze pas cu pas îndeplinirea programului în baza exemplurilor concrete;
- să analizeze corectitudinea aplicării uneia sau alteia structuri de control în program.

### Conținuturi

#### a) Prelegeri

Nr. d/o	Subiectele de studiu	Nr. ore
<b>Unitatea de conținut 1: Structura unui program. Instrucțiuni. Tipuri de date simple, omogene – 15 ore</b>		
1.	Generalități privind limbajele de programare. Medii de programare. Sintaxa și alfabetul limbajului. Cuvintele cheie. Blocuri de instrucțiuni.	1
2.	Tipuri de date. Constante și variabile. Declararea variabilelor. Adresele și inițializarea variabilelor.	1
3.	Funcțiile de citire și de scriere a datelor în mod standard.	1
4.	Operatori: aritmetici, de atribuire, de incrementare și decrementare, relaționali, logici, la nivel de bit, condițional, de rezoluție.	1
5.	Instrucțiuni de decizie: <i>if, if-else, switch</i> .	1
6.	Instrucțiuni de ciclare: <i>for, while, do-while</i> . Clasificarea ciclurilor. Instrucțiunile <i>break</i> și <i>continue, goto, exit</i> .	2
7.	Definirea și declararea funcțiilor. Apelul funcțiilor. Parametri formali și actuali. Transmiterea parametrilor prin: valoare, adresă, referință. Variabile globale și locale. Variabile automate, statice, externe. Durata de viață a variabilelor.	2
8.	Directive de preprocesare: <i>#define, #include, #if, #ifdef, #ifndef, #endif, #else</i> .	1
9.	Pointeri. Declararea și utilizarea pointerilor. Utilizarea pointerilor ca parametri ai funcțiilor. Mecanismul de indirectare.	1
10.	Tabele. Tabele unidimensionale. Inițializarea tabelului. Accesarea elementelor unui vector. Utilizarea tabelului ca pointeri. Algoritmul de căutare în vector. Determinarea valorii minime și maxime. Algoritm de formare a unui vector din alți vectori. Algoritmul de deplasare a elementelor unui vector. Algoritmi de rotire a elementelor unui vector.	2

	Algoritmi de sortare a vectorilor.	
11.	Tabele bidimensionale. Accesul la elementele tabelului. Prelucrarea tabelelor pe rânduri sau coloane. Algoritmi de formare a tabelelor. Determinarea elementelor minime și maxime. Includerea unui rând / unei coloane în tabel. Excluderea unui rând sau a unei coloane din tabel. Generarea matricelor.	2
<b>Unitatea de conținut 2: Șiruri de caractere. Tipuri de date derivate. Apeluri recursive - 12 ore</b>		
12.	Șiruri de caractere. Utilizarea șirurilor de caractere ca pointeri. Funcții standard de prelucrare a șirurilor de caractere.	2
13.	Structuri și uniuni. Accesarea elementelor structurii. Structuri înlănțuite. Inițializarea structurilor. Redenumirea tipurilor de date. Pointeri la structuri. Câmpuri. Tipul enumerare.	2
14.	Fișiere. Modul standard de intrare-ieșire. Clasificarea fișierelor. Deschiderea și închiderea unui fișier. Modul de acces la deschiderea fișierelor. Scrierea și citirea fișierelor de tipul: caracter, șir de caracter, formatat, înregistrare. Accesarea directă și secvențială a articolelor unui fișier. Fișiere standard.	2
15.	Recursia. Recursie directă și indirectă.	1
16.	Supraîncărcarea funcțiilor. Funcții inline. Funcții cu parametri implicați.	1
17.	Structuri dinamice de date. Stiva. Coadă. Lista liniară. Liste dublu înlănțuite și liste dublu înlănțuite cu santinele. Implementarea stivelor, cozilor, listelor. Operațiile posibile cu o structurile dinamice de date.	2
18.	Elaborarea programelor complexe. Fișiere proiect. Etapele de realizare a unui program complex. Modele de memorie. Tipurile de pointeri.	2
<b>Unitatea de conținut 3: Elemente de programare orientată pe obiecte - 3 ore</b>		
19.	Elemente fundamentale ale POO. Clase și obiecte. Modificatorii de protecție: private, protected și public. Mesaje și metode. Constructori. Destructorii. Funcții prietene. Operatori supradefiniți și operatori prieteni.	2
20.	Moștenirea. Polimorfismul. Ierarhii de clase. Funcții virtuale. Supradefinirea funcțiilor membre și a operatorilor. Clase prietene. Moștenirea multiplă. Clase abstracte.	1
<b>Total</b>		<b>30</b>

a) **Laborator.** La orele de laborator sunt realizate 5 proiecte de program (4 proiecte – de bază, iar 1 – opțional) și o lecție de inițiere.

#### **Tematica proiectelor de program și repartizarea orelor**

<b>Nr. D/o</b>	<b>Tema</b>	<b>Nr. de ore</b>
1.	Mediul de programare. Înscrierea programelor. Compilarea. Link - editarea. Lansarea în execuție. Instrucțiuni de intrare – ieșire. Trasarea pe pași a programului. Utilizare Help.	2
2.	Proiect de programe Nr 1. Aplicarea structurilor de control în program. Algoritmi clasici.	4
3.	Proiect de programe nr. 2. Gestiunea tabelelor de numere. Pointeri. Funcții.	6
4.	Proiect de programe nr. 3. Șiruri de caractere. Fișiere text.	6
5.	Proiect de programe nr. 4. Structuri. Fișiere binare. Structuri dinamice.	6
6.	Proiect de programe nr. 5. Fișiere proiect. Clase. (opțional)	4
7.	Evaluare	2
<b>Total</b>		<b>30</b>

Fiecare student îndeplinește un set de acțiuni, conform setului specificat de proiecte de program, unde sunt indicate: scopul lucrării; formularea problemei; datele inițiale; indicații metodice și recomandări; cerințele față de program; variantele însărcinărilor.

### **Activități de lucru independent**

În cadrul acestei unități de curs, studenților li se propun câte două sarcini, care urmează a fi studiate și rezolvate independent.

Sarcinile pentru activitate independentă sunt propuse la finele studiului primei și celei de-a doua unități de învățare. În special, sunt reflectate sarcini de implementare a materialului teoretic cu aplicații în practică.

Consultațiile pentru lucru independent sunt afișate în orarul catedrei.

*Sarcina nr. 1 pentru lucrul independent:* Aplicarea tipurilor de date omogene la rezolvarea unor probleme complexe concrete. Exemplu:

1. De determinat nota maximă (minimă) a studenților la o unitate de curs, notele fiind păstrate într-un tabel unidimensional.
2. În continuare, de determinat nota maximă (minimă) a studenților la mai multe unități de curs, notele fiind păstrate într-un tabel bidimensional. De făcut legătura cu un tabel unidimensional, în care sunt introduse numele studenților.

**Termenul limită de prezentare a sarcinii nr. 1 – 15 aprilie.**

*Sarcina nr. 2 pentru lucrul independent:* Aplicarea tipurilor de date mixte la rezolvarea a unor probleme complexe concrete. Exemplu:

1. În continuare, la sarcina nr. 2, datele se citesc din fișier text, apoi binar.
2. În final, datele se protejează, folosind clasele și conceptele programării orientate pe obiect.

**Termenul limită de prezentare a sarcinii nr. 2 – 22 mai.**

### **Strategii didactice utilizate**

Expunerea, problematizarea, prelegerea interactivă, demonstrația, explicația, conversația euristică.

### **Strategii didactice utilizate**

Expunerea, problematizarea, prelegerea interactivă, demonstrația, explicația, conversația euristică.

## Evaluarea

Evaluarea studenților se realizează în corespundere cu Regulamentul cu privire la evaluarea rezultatelor academice ale studenților în Universitatea de Stat „Alec Russo” din Bălți, aprobat prin Hotărârea Senatului Universității de Stat „Alec Russo” din Bălți (procesul-verbal nr. 9 din 16.03.2011).

Cunoștințele, capacitățile și competențele studenților vor fi evaluate:

- în cadrul orelor de curs prin realizarea a 2 lucrări de control. De asemenea studenții pot fi evaluați la orele de prelegeri prin rezolvări de probleme, argumentări a elaborării algoritmului de calcul. Prima lucrare de control este evaluată pe platforma MOODLE la calculator în formă de test pe unitatea de conținut nr. 1. A doua lucrare de control este realizată în scris, în formă de test cu 8 itemi și 2 sarcini practice de definire a unei funcții pe unitatea de conținut nr. 2 și ultimele teme din unitatea de conținut nr. 1;
- în cadrul orelor practice/lucrărilor de laborator, prin îndeplinire a celor 4 proiecte de program pentru fiecare unitate de conținut. Fiecărui student îi este prezentat modul de evaluare a proiectelor (conform calendarului disciplinei);
- prin realizarea a 3 sarcini de lucru independent (conform calendarului disciplinei);
- la examenul final.

Nota finală la disciplina „Informatica II” se calculează conform formulei:

$$N_f = 0,6 \times n_c + 0,4 \times n_e,$$

unde  $N_f$  – nota finală;  $n_c$  – media curentă,  $n_e$  – nota de la examen;

$$n_{l_i} = (s_{i_1} + s_{i_2})/2$$

$$m_{lab} = (note_{proiecte} + n_{l_i})/nr_{note}$$

$$n_c = (m_{lab} + m_{curs}) / 2$$

unde  $n_c$  – media curentă,  $m_{lab}$  – media notelor pentru proiectele de program,  $m_{curs}$  – media notelor pentru evaluările curente de la orele de curs,  $n_{l_i}$  – media notelor pentru sarcinile de lucru independent,  $m_{curs}$  – media notelor pentru evaluările curente de la orele de curs,  $s_{i_1}$  și  $s_{i_2}$  – note pentru sarcinile de lucru independent.

Examenul final se susține în scris. Lista întrebărilor pentru examen include doar temele indicate în conținuturi.



## Resurse informaționale la unitatea de curs

### Obligatorie

1. CHERCEZ, E; ȘERBAN, M. *Programarea în limbajul C/C++ pentru liceu*. Vol. I. Iași, Polirom, 2005.
2. CHERCEZ, E; ȘERBAN, M. *Programarea în limbajul C/C++ pentru liceu*. Vol. II. Iași, Polirom, 2005.
3. CHERCEZ, E; ȘERBAN, M. *Programarea în limbajul C/C++ pentru liceu*. Vol. III. Iași, Polirom, 20010.
4. NEGRESCU, L. *Limbajele C și C++ pentru începători. Volumul 1. Limbajul C. Volumul 2. Limbajul C++*. Cluj, 1996.
5. BARKAKATI, N. *Borland C++. Ghidul programatorului*. București: Teora, 1997.
6. BALTAC, I. *Programarea în limbajele C și C++*. București, Teora, 1996.
7. NAMIR, C. *Curs rapid de Borland C++ 4*. București: Teora, 1996.
8. СЕДЖВИК, С. *Язык программирования Си++*. М.: Мир, 2001.
9. СТРАУСТРУП, Б. *Язык программирования C++*. 3-е изд. СПб.: Невский Диалект, 1999.
10. ПОЛ, А. *Объектно-ориентированное программирование на C++*. СПб.: Питер, 2001.

### Suplimentară:

11. RÎNGACI, V. D., etc. *Programare în limbajul Turbo C. (în 3 părți)*. Chișinău, 1994.
12. БАДД, Т. *Объектно-ориентированное программирование в действии*. СПб.: Питер, 1997.

## Principiile de lucru în cadrul unității de curs

1. O parte din sarcinile de învățare vor fi propuse pentru realizare în grupuri mici, prin cooperare. Deși activitatea de învățare va fi una colectivă, notele pentru realizarea sarcinilor vor fi individuale.
2. Calendarul cursului este corelat cu calendarele altor unități de curs din semestru. De aceea, prezentarea sarcinilor după termenul limită indicat în calendar nu este salutăată, iar studenții care amână frecvent prezentarea sarcinilor își formează o imagine nefavorabilă.
3. Nu este salutăată întârzierea la ore.
4. Este salutăată poziția activă a studentului, care studiază din propria inițiativă noi conținuturi, propune soluții, formulează întrebări în cadrul prelegerilor și orelor practice.
5. În cadrul unității de curs, cu atenție sporită, se vor respecta principiile *etice*. Prezentarea unor soluții a sarcinilor, preluate de la colegi sau din alte surse, preluarea informațiilor din diverse surse, fără a face trimitere la sursă, va fi considerată plagiat și va fi sancționată prin note de „1”.